AE598: Homework 1 - Model-based and Model-Free learning methods for Gridworld and Inverted Pendulum

Niket Parikh *
University of Illinois Urbana-Champaign, Urbana, Illinois, 61801

This report summarizes the problem solved in the HW1, mentions the specific implementation details needed to understand the algorithm implementations and presents the results followed by a brief discussion.

I. Introduction

This report encapsulates the work done for HW1 and presents the results. The aim of the work is to implement five reinforcement learning (RL) algorithms, namely, policy iteration (PI), value iteration (VI), SARSA and Q-learning on two distinct environments. The first environment is a simple 5×5 gridworld with four actions, which are up, left, down, right. The state transitions are deterministic and the transition model is known. Two specific state transitions are rewarded highly while all other transition have zero reward except those which attempt to move out of the grid, which results in a small negative reward. The second environment is a discretized (both state space and action space) simple pendulum for which an explicit transition model is not known. The actions consist of torque values that can be applied to the pendulum mass while the state is an integer obtained from the tuple of pendulum angle and angular velocity. The agent is rewarded with a small positive value for staying upright between a specific threshold of pendulum angles, a high negative reward for exceeding an angular velocity threshold and zero reward otherwise.

II. Problem Setup

The problem is expressed as a Markov Decision Process with infinite horizon and discount factor, $\gamma = 0.95$.

III. Details of Implementation

This section highlights the design choices (such as hyperparameter values) made during implementation of the algorithms. The implementation for each algorithm in unchanged for the environments. The value of discretising parameters for the pendulum environment are $n_\theta = 15, \dot{n}_\theta = 21$ and $n_actions = 31$.

A. Policy Iteration

The implementation is standard. The state-value function is initialised by sampling uniformly from the interval $[-100, 100)$ instead of zero values. In-place policy evaluation is used. Policy improvement is done using a greedy approach. The threshold parameter, θ for breaking out of policy evaluation when error less than threshold error has been achieved, is $\theta = 0.0005$.

B. Value Iteration

The implementation is standard. The state-value function is initialised by sampling uniformly from the interval $[-100, 100)$ instead of zero values. In-place updates on the value function are performed. The threshold parameter, θ for breaking out of policy evaluation when error less than threshold error has been achieved, is $\theta = 0.0005$.

C. SARSA

The implementation is standard. The state-action value function is initialised with zero values. Epsilon-greedy policies are used to sample the actions during training. The optimal policy is greedy with respect to the final action value function that the algorithm converges to.

The value of epsilon-greedy's parameter, ϵ, is 0.3 for all the plots in this report except where noted otherwise. The value of the averaging parameter, α, during the value updates, is 0.1 for all the plots in this report except where noted

*PhD Student, Aerospace Engineering, University of Illinois Urbana-Champaign

otherwise. For the gridworld, the training is carried out for 5000 episodes. For the pendulum, the training is carried out for 7000 episodes.

D. Q-Learning

The implementation is standard. The state-action value function is initialised with zero values. Epsilon-greedy policies are used to sample the action at the current state during training. The optimal policy is greedy with respect to the final action value function that the algorithm converges to.

The value of epsilon-greedy's parameter, ϵ, is 0.3 for all the plots in this report except where noted otherwise. The value of the averaging parameter, α, during the value updates, is 0.1 for all the plots in this report except where noted otherwise. For the gridworld, the training is carried out for 5000 episodes. For the pendulum, the training is carried out for 7000 episodes.

E. TD(0)

The implementation is standard. Note that the algorithm uses the same parameter values, α and number of episodes for training, as those for the algorithm whose optimal policy is being evaluated.

IV. Results

Please refer to the previous section for the value of parameters used to obtain these plots. The number of plots are as required by the homework description.

The pendulum angle is wrapped to stay between $[-\pi, \pi]$. For varying ϵ plots, the values investigated are $0.1, 0.3, 0.5, 0.7, 0.9$ and for varying α plots, the the values investigated are $0.001, 0.01, 0.1, 0.3, 0.5$. It's been observed that $\epsilon = 0.3$ gives the best results for gridworld and one of the better ones for pendulum (along with $\epsilon - 0.1$). For α, it is again the same values that give the best or one of the better results.

The algorithms work well for gridworld and the state trajectories of trained agent are optimal. For pendulum however, the results are not very satisfactory since the pendulum doesn't stay upright for long enough (it does for some intervals, particularly for SARSA agent). This primarily might be due to discretization not being good enough (more number of grids might be needed) or much larger number of episodes required for training (7000 episodes might not be enough).

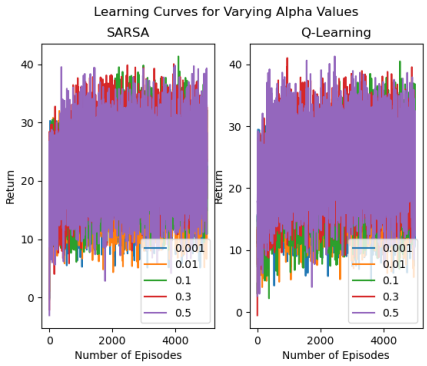

Fig. 1 Learning curves for varying α values for the model-free methods applied to gridworld environment

Acknowledgments

The author is grateful to the course instructor, Prof. Huy Tran, for providing codes for the environment Python classes.

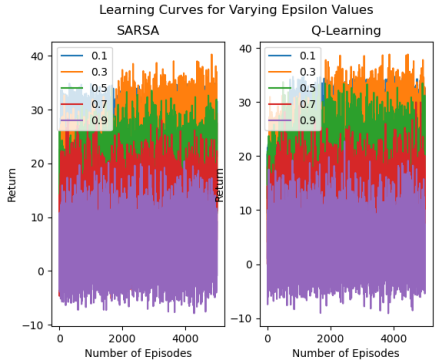

Fig. 2 Learning curves for varying ϵ values for the model-free methods applied to gridworld environment

Fig. 3 Action chosen by the deterministic policy of the trained agent for all the methods applied to gridworld environment

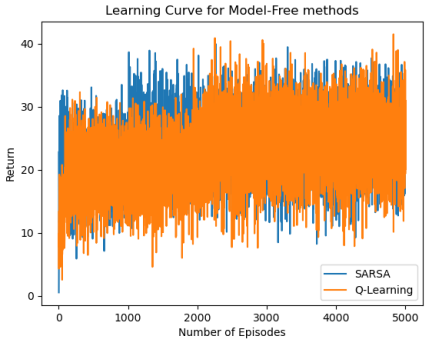

Fig. 4 Learning curves for the model-free methods applied to gridworld environment. The ϵ and α values used are same as for other plots except for plots where the preceding two parameters are varied

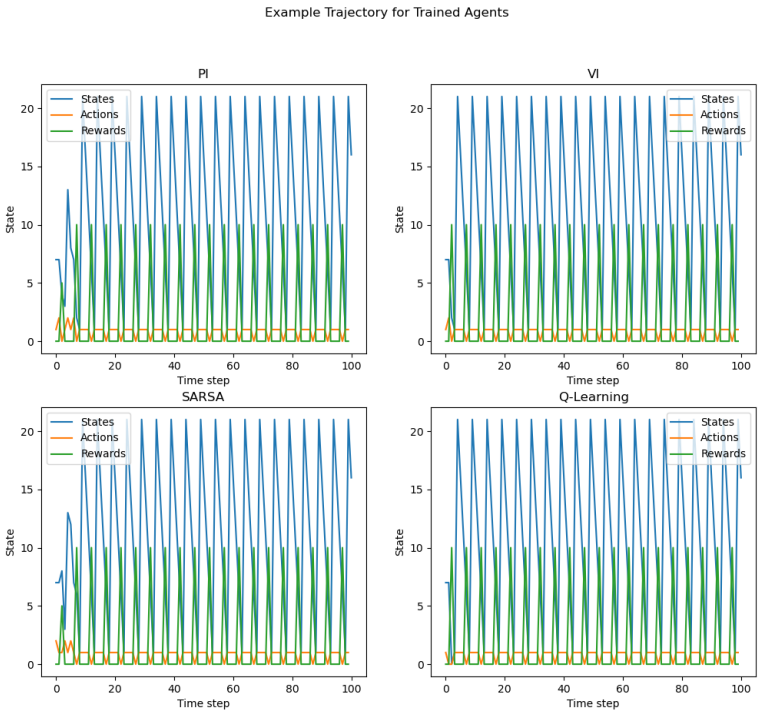

Fig. 5 State, action and reward trajectories of trained agents for all the methods applied to gridworld environment

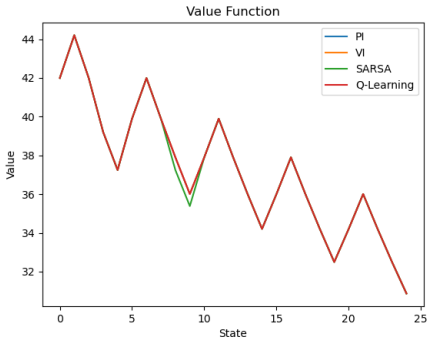

Fig. 6 Value for each state as obtained from all the methods applied to gridworld environment

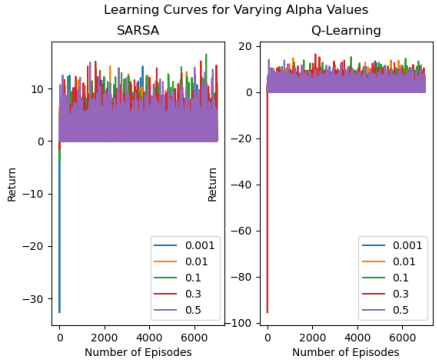

Fig. 7 Learning curves for varying α values for the model-free methods applied to pendulum environment

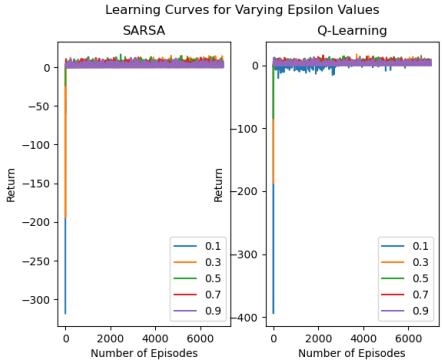

Fig. 8 Learning curves for varying ϵ values for the model-free methods applied to pendulum environment

Fig. 9 Action chosen by the deterministic policy of the trained agent for all the methods applied to pendulum environment

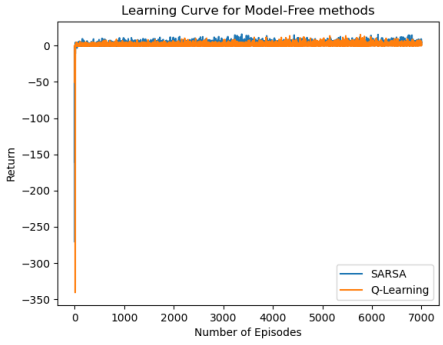

Fig. 10 Learning curves for the model-free methods applied to gridworld environment. The ϵ and α values used are same as for other plots except for plots where the preceding two parameters are varied

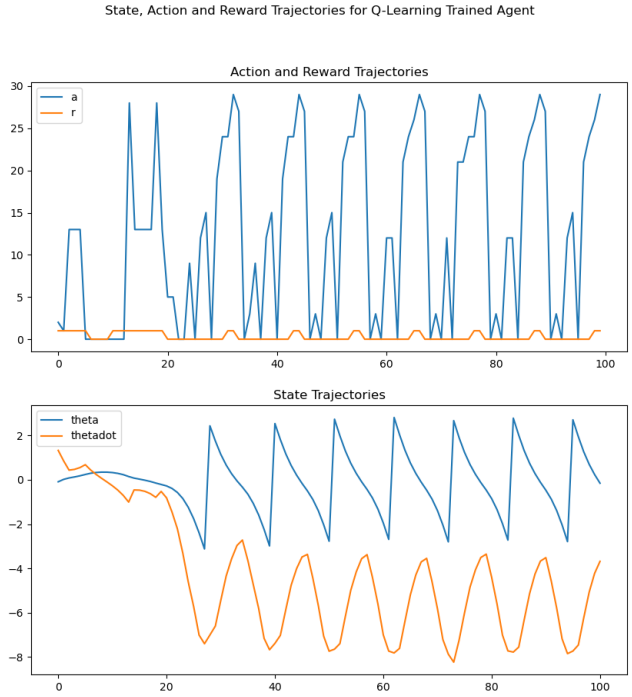

Fig. 11 State, action and reward trajectories of trained agents for Q-Learning applied to pendulum environment

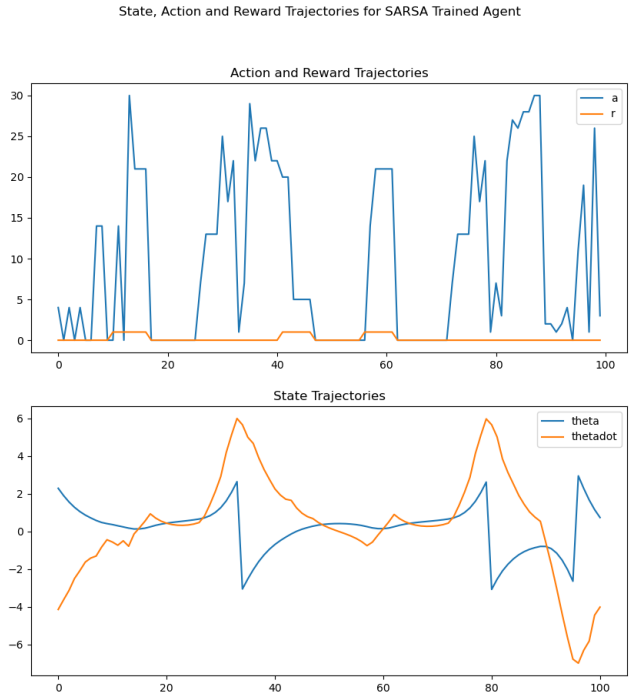

Fig. 12 State, action and reward trajectories of trained agents for SARSA applied to pendulum environment

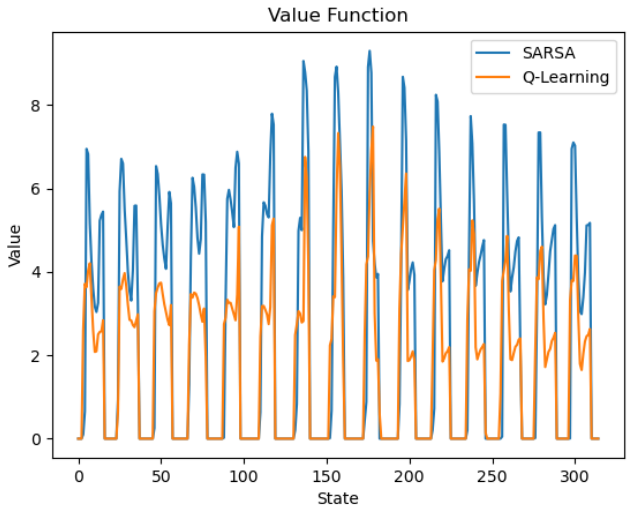

Fig. 13 Value for each state as obtained from all the methods applied to pendulum environment